DEVELOPMENT OF OPEN-SOURCE SOFTWARE FOR ARABIC TEXT STEMMING AND CLASSIFICATION

Ashrf Ali Nasef,
Marina Marjanović-Jakovljević

Singidunum University,
32 Danijelova Street, Belgrade, Serbia

Abstract:
This paper presents implementation of a software tool for Arabic text stemming and classification. The software tool is based on open source implementation of the Lucene based Light stemmer for Arabic language and it enables stemming and classification into 12 categories (religion, economics, sports, science, computers, history, medicine, entertainment, engineering, literature, politics and food).

Key words:
arabic language, text mining, stemming, text classification, open source code.

1. INTRODUCTION AND THEORETICAL BACKGROUND

The Arabic language is the seventh most frequently used language worldwide. There are more than 310 million native Arabic speakers scattered throughout the world.

It is considered that over 38 million Arabic people use the Internet at least once a month, while the total Internet penetration (the relation between the overall population and the users of the Internet) extended to 11.1% [1]. This number presents just half of the global average (21.9%). Nevertheless, there are indicators showing that this number will grow rapidly. The largest number of Internet users comes from Egypt (8.6 million), then Morocco (7.3 million) as well as Saudi Arabia (6.2 million). Concerning Libya, Internet penetration was rather low, *e.g.* in 2010 it was slightly below 14% (13.77%) [2]. But, the growth of Internet users in Libya has almost an exponential trend for the period 1999-2010 [1].

It wasn't until the early 90's when the Arabic IR research developed. The representation of Arabic text and coding were in the center of the researchers' attention prior to that. There has been a certain number of studies about various methodologies of integrating morphology: root, stem, light stem and no-stemming and in addition to that, the use of non-rule based statistical models or n-gram models.

The analysis of Arabic text mining and Arabic IR research has shown that, due to language specifics, Arabic language text mining task can be quite challenging compared to other languages. Even though extensive research activities regarding this subject matter have been performed, which results in significant achievements, there is still a huge amount of work to be done in order to align Arabic language IR and text mining tools with

Correspondence:
Ashrf Ali Nasef

e-mail:
ashraf8259@yahoo.com

DOI: 10.15308/Sinteza-2016-271-276

those existing for Western widely spoken languages such as English, German, French and Spanish. Arabic language stemming is a task that requires research as well Arabic text classification. Also, open source software tools/components for Arabic text mining are at infancy stage, as can be seen from the most prominent relevant open source project "*Arabic Computational Linguistics*" [3].

Concerning the rapid expansion of the Internet and the fact that it's mostly used nowadays to search for information, there is a need to develop tools for Arabic text mining. Since the Arabic language possesses a complex morphological structure, this will be a demanding task as the next section will show.

2. ARABIC LANGUAGE AND ARABIC STEMMERS

Arabic language

Arabic letters

Words in the Arabic language are constituted of twenty-nine constant letters. Arabic letters are classified into two types: letters of signification and letters of construction. We use letters of signification for sentence formation and modification of nouns and verbs, while letters of construction are used for word formation. Arabic prefixes can be formed of letters of signification and they can be one letter or a combination of two or more. Additionally, the shape of most letters is determined by their position in a sentence, as well as by the adjacent letters.

Arabic Affixes

Affixes in the Arabic language could be a letter or a letter combination. For example, certain suffixes are added to words in order to change their form from singular to plural, while other suffixes are added to imply masculine or feminine gender. By adding the affixes to a noun, we can get 1440 different verb forms. Affixes are classified according to their position in the word: prefixes are added to the beginning of the word, suffixes to the end of the word, while infixes are added into the middle of the word.

Arabic vowels

The Arabic language has vowels (short, long and double) and consonants, but unlike in English, in the Arabic language, the main (short) vowels are represented by a symbol and not by a letter. The symbols are added below or above the text. Short vowels in the Arabic language are very powerful. They determine the pronunciation of the word and can also change its meaning. In Arabic modern writing, these short vowels are omitted so Arabic writing can be vocalized or unvocalized. This omission of short vowels causes problems when the words are placed into a certain context and this is a large barrier for Arabic text processing.

Arabic root base system

The Arabic language is a Semitic language with a root system foundation. Unlike English, roots in Arabic can only be verbs. These roots are mostly 3 literal words and it is supposed that there are about 10000 independent roots in the Arabic language. Regular and irregular tenses, nouns or the adjectives are created by adding suffixes, infixes or prefixes. Words that originate from the same root don't have to be related semantically, whereas one root can also have various meanings.

Nouns and pronouns

In Arabic, a noun is derived from the root by applying specific patterns. Nouns in the Arabic language can at the same time possess a regular and irregular plural, which means that Arabic nouns have duals. Nouns in the Arabic language possess either feminine or masculine gender. The Arabic language pronouns are classified into: personal pronouns, demonstrative pronouns, possessive pronouns, and relative pronouns. We use personal pronouns instead of nouns. Demonstrative pronouns are analogous to this/that and these/those. Relative pronouns are analogous to English who, which, what *etc.* and as far as Arabic text processing is concerned they are considered to be stop words. Alternatively, possessive and object pronouns demonstrate that some pronouns are word suffixes. Possessive pronouns are added to nouns to refer to possession, or to turn them into definite, while object pronouns are added to verbs.

Arabic Stemmers

According to [5]: "As far as the information retrieval and linguistic morphology are concerned, stemming is the activity of reduction of inflected (or derived in some

cases) words to their base, root or stem - typically the written form of a word."

There are two different categories of Arabic stemmers according to the necessary level of analysis. They are classified into stem-based or root-based stemmers. A morpheme or a combination of linked morphemes to which an affix can be added is called a stem. On the other hand, a root is the initial word form without any alteration process. Khoja's stemmer represents the most famous Arabic stemmer. In addition to that, Light stemmers with light 1, 2, 3, 8 and 10 stemmers are also widely known.

Khoja's Stemmer

The Khoja algorithm is used for the removal of suffixes, prefixes and infixes, whereas pattern matching is used to isolate roots from the dictionary. Khoja's stemmer algorithm as presented in [4] consists of the following steps:

1. Replace initial آ أ إ with ا
2. Eliminate stop words.
3. Eliminate punctuation, diacritics and non-letter forms.
4. Eliminate definite articles from word beginning.
5. Eliminate the letter (و) from the word beginning and (ة) from the word end
6. Eliminate prefixes and suffixes
7. Make a comparison between the words in the dictionary and the obtained word. If the obtained root doesn't have a meaning, the initial word is returned without any modifications

Certain problems are present in the algorithm, particularly regarding proper nouns or broken plurals. Furthermore, the dictionary needed to be maintained regularly to add new words.

Light stemmers

Light stemmers containing light 1, 2, 3, 8 and 10 have shown to outperform previous light-stemmers as well as Khoja's root based stemmer [4]. Light stemmers don't cope with infixes or certain patterns. It simply removes prefixes and/or suffixes. Larkey's light 10 stemmer algorithm [7] as presented in [4] consists of the following steps:

1. Eliminate stop words,
2. Eliminate punctuation, diacritics and non-letter forms, as well as the non-Arabic forms,
3. Initial آ أ إ is replaced with ا
4. Switch ى to ي and ة to ه
5. Eliminate the letter و from the word beginning just in case that the obtained word has more than three letters,
6. Eliminate definite articles ال. كال. فال. وال. بال from the word beginning just in case that the obtained word has two or more than two letters,
7. Eliminate the suffixes ي . ة . ه . تي . هي . ني . نو . تا . ان . اه from the word end (longer one first) just in case that the obtained word has two or more than two letters.

Many suffixes are ignored by the Larkey's light stemmer, which can be the reason for a high rate of understemming errors. Moreover, the suffix and prefix list could be original letters.

3. OPEN SOURCE BASED ARABIC STEMMING AND CLASIFICATION TOOL

Using the Lucene based Arabic light stemmer from Arabic Computational Linguistics project [3], we've constructed an Arabic stemming tool that helped us to develop the Arabic language classifier tool additionally.

In this work, a brief outline of the idea will be presented.

Arabic stemming tool prompts the user for a file containing Arabic text and then stems and displays all words included in the text. The program consists of three parts: graphical interface, file parser and the stemmer.

Graphical interface consists of the initial dialog used to browse a file and start the stemming and the results display window. The initial dialog provides for selecting the file containing the text to be stemmed.

File parser is a class that takes the chosen file and parses its content into an array of words and turns the content of a file into an array of strings. These strings will be stemmed later on. Line by line of a text form of the file is read by the class, whereas the StringTokenizer class is used to separate words. The characters that are excluded from the final results are : .,!?;:"() and all white spaces.

The stemmer consists of Arabic normalizer, Arabic stemmer and utilities for processing character arrays.

The architecture of the program is shown in Figure 1.

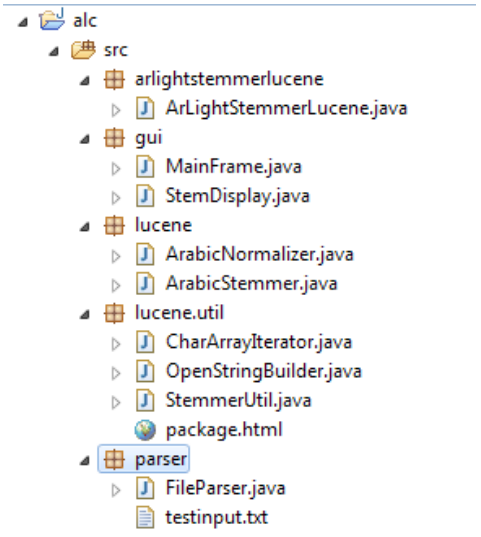

Figure 1. The architecture of the Arabic stemmer

Two screen shots illustrating user interface are presented in Figures 2 and 3.

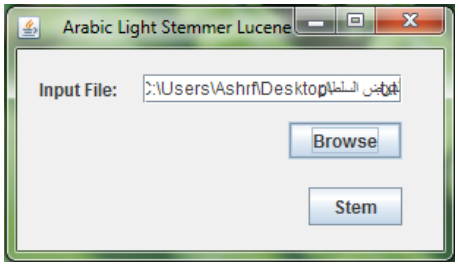

Figure 2. Graphical interface for selecting file to be stemmed

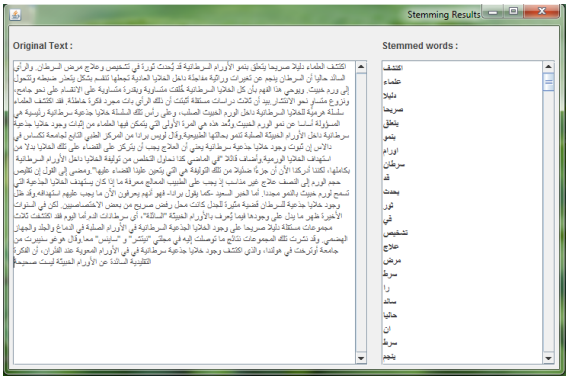

Figure 3. Graphical interface for displaying stemming results

Classification tool

The classifier prompts the user to input a folder and classifies Arabic files from the selected folder into 12 categories: religion, economics, sports, science, computers, history, medicine, entertainment, engineering, literature, politics and food. The classifier firstly stems the words from each individual document and then finds two keywords in each document. Using the database dictionary where all words are tagged, documents can be classified based on their key words. The classifier also has a training mode which allows the user to tag newly found words.

The architecture of Arabic classifier tool is depicted in Figure 4.

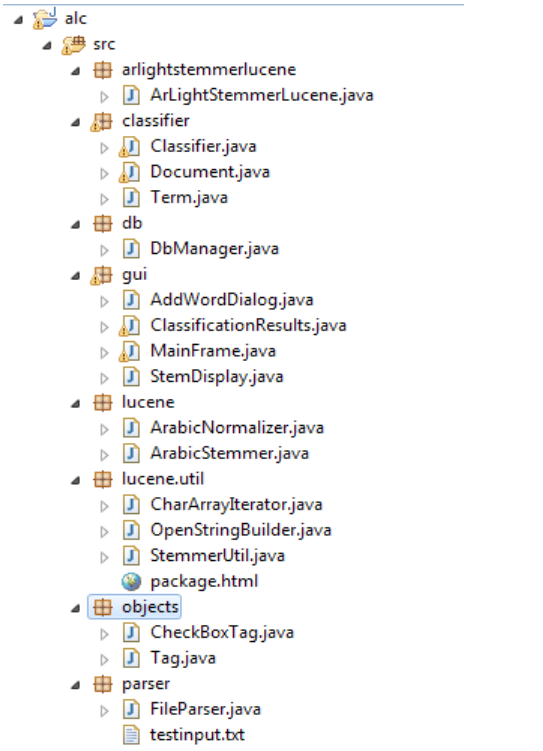

Fig. 4. The architecture of the Arabic classifier

The classifier consists of 5 parts: the stemmer and the file parser both described in the previous chapter, the interface to the database, the graphical interface for prompting and displaying the results and the classifier.

The interface to the database is in the file DbManager.java, a singleton with the methods for adding and selecting items from the database.

Added classes to the graphical interface are MainFrame.java which is the opening dialog, AddWordDialog.java which is the dialog for tagging newly found words and the class which is a frame for displaying results- ClassificationResults.java.

The Classifier.java class is the class that does the actual classification. Important classes for classification are Document.java and Term.java. These classes abstract the document and the term in text mining.

Two screen shots illustrating user interface are presented in Figures 5, 6 and 7.

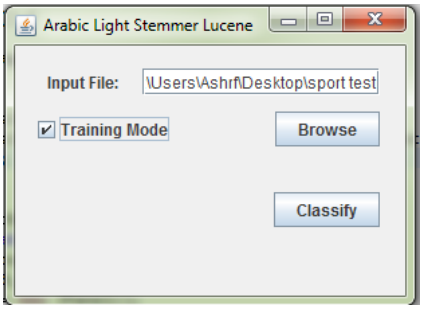

Figure 5. Graphical Interface for Choosing the folder containing the File to be Classified or used for Training

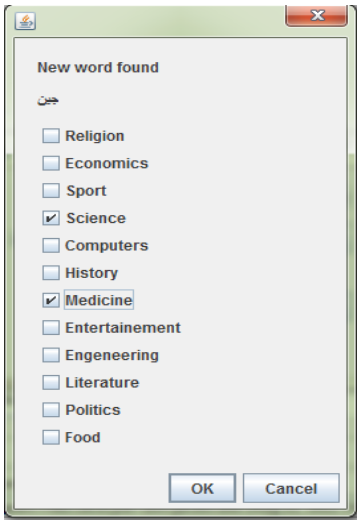

Figure 6. Graphical Interface for Tagging words in the Training Mode

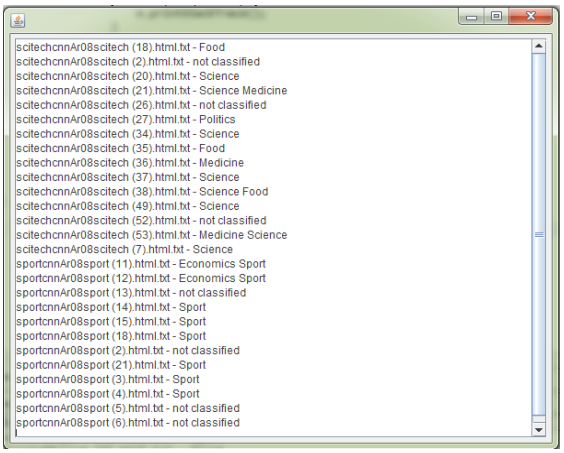

Figure 7. Graphical Interface for Display Results

4. CONCLUSION AND FUTURE WORK

The aim of this paper was to present the author's work on developing software and graphical interface for Arabic text stemming and classification.

Based on the previous analysis of literature sources, the Internet as searched for available open source solutions aimed at Arabic text mining, and suitable (most complete/mature solution) open source software components for constructing software tool for Arabic text stemming and classification were selected and integrated to form a tool prototype. Finally, the prototype was tested against planned functionalities.

Based on the analysis concerning Arabic text mining and Arabic IR research, it can be concluded that Arabic language text mining can be quite difficult compared to other languages, mainly due to language specifics. Even though extensive research activities regarding this subject matter exist, which results in significant achievements, there is still a huge amount of work to be done in order to align Arabic language IR and text mining tools with those existing for Western widely spoken languages such as English, German, French and Spanish. The Arabic language stemming is one task that requires further research as well as Arabic text classification. Also, open source software tools/components for Arabic text mining are at infancy stage, as can be seen from the most prominent relevant open source project "Arabic Computational Linguistics".

Taking into account the results achieved so far and conclusions derived by this master thesis, further research could be divided into two directions. One line is further research concerning Arabic computational linguistics and Arabic text mining (developing different approaches of incorporating language morphology), while the other line is development of open source software tools/components aimed at Arabic text mining and their applications to various fields such as spam filtering, creating suggestion and recommendations, monitoring public opinions, *etc.*

REFERENCES

[1] The Internet in Arab countries al-bab. Available from: http://www.al-bab.com/media/Internet.htm [Accessed 10/03/16].

[2] Internet Users IndexMundi. Available from: http://www.indexmundi.com/facts/indicators/IT.NET.USER.P2 [Accessed 10/03/16].

[3] Arabic Computational Linguistics project ("ar-text-mining") SourceForge.net. Available from: http://ar-text-mining.sourceforge.net/ [Accessed 10/03/16].

[4] Al-Shammari, E.T., Lin, J. Towards an error free Arabic stemming, iNEWS '08 Proceedings of the 2nd ACM workshop on improving non English web searching, New York: ACM New York, 2008, pp. 9-16.

[5] Wikimedia project Stemming Wikimedia Foundation, Inc. Available from: http://en.wikipedia.org/wiki/Stemming. [Accessed 10/03/16].

[6] Nasef, A. An Open Source based Software Tool for Arabic Text Stemming and Classification. Unpublished MSc thesis, University of Novi Sad Faculty of Technical Scineces, 2012.

[7] Larkey L.S., Ballesteros L. and Connell M.E. (2007) Light Stemming for Arabic Information Retrieval. In: SOUDI, A. *et al.* (eds.) Arabic Computational Morphology. 1st ed. Berlin: Springer, pp. 221 - 243.